

Assessing Quality Metrics for Neural Reality Gap Input Mitigation in Autonomous Driving Testing

Stefano Carlo Lambertenghi

Technical University of Munich, fortiss GmbH

Munich, Germany

stefanocarlo.lambertenghi@tum.de, lambertenghi@fortiss.org

Andrea Stocco

Technical University of Munich, fortiss GmbH

Munich, Germany

andrea.stocco@tum.de, stocco@fortiss.org

Abstract—Simulation-based testing of automated driving systems (ADS) is the industry standard, being a controlled, safe, and cost-effective alternative to real-world testing. Despite these advantages, virtual simulations often fail to accurately replicate real-world conditions like image fidelity, texture representation, and environmental accuracy. This can lead to significant differences in ADS behavior between simulated and real-world domains, a phenomenon known as the sim2real gap.

Researchers have used Image-to-Image (I2I) neural translation to mitigate the sim2real gap, enhancing the realism of simulated environments by transforming synthetic data into more authentic representations of real-world conditions. However, while promising, these techniques may potentially introduce artifacts, distortions, or inconsistencies in the generated data that can affect the effectiveness of ADS testing.

In our empirical study, we investigated how the quality of image-to-image (I2I) techniques influences the mitigation of the sim2real gap, using a set of established metrics from the literature. We evaluated two popular generative I2I architectures, pix2pix and CycleGAN, across two ADS perception tasks at a model level, namely vehicle detection and end-to-end lane keeping, using paired simulated and real-world datasets. Our findings reveal that the effectiveness of I2I architectures varies across different ADS tasks, and existing evaluation metrics do not consistently align with the ADS behavior. Thus, we conducted task-specific fine-tuning of perception metrics, which yielded a stronger correlation. Our findings indicate that a perception metric that incorporates semantic elements, tailored to each task, can facilitate selecting the most appropriate I2I technique for a reliable assessment of the sim2real gap mitigation.

Index Terms—autonomous vehicles testing, generative adversarial networks, sim2real, reality gap.

I. INTRODUCTION

Autonomous driving systems (ADS) are vehicles capable of traveling between destinations with little to no human intervention. Current ADS are equipped with advanced deep neural networks (DNNs) that process sensor data to perform different prediction and control tasks, such as lane keeping, object detection, object avoidance, and path planning [1], [2].

To validate the safety of ADS, the consolidated industrial practice by automakers includes collecting real-world driving data that is used within a multi-pillar approach that includes training driving policies, virtual testing, test track, and real-world testing [3]. Virtual testing uses simulation toolkits to

assess the compliance of an ADS with the safety requirements on a wide range of virtual scenarios, some of which are extremely difficult to test in the real world [4], [5].

However, generalization to real-world conditions is not guaranteed [6], as simulation platforms typically lack sensor fidelity and accurate representation of real-world conditions. This phenomenon, known as the reality gap or sim2real gap [2], [7], causes a mismatch between the behavior of autonomous vehicles in simulated environments and the real world. For ADS, understanding and mitigating the sim2real gap is crucial. When the ADS performance in the simulated environment does not directly translate to its performance on actual roads, it lowers the trustworthiness of testing and acceptance of ADS for large-scale deployment.

In this work, we target the sim2real gap for vision-based ADS at a model level, i.e., considering pairs of simulated and real-world inputs to the ADS. In this context, data-driven neural solutions, specifically Image-to-Image translation models (referred to as I2I models hereafter), are used to transform input images pertaining to one domain (i.e., simulated driving scenes) into new images in a different domain (i.e., real-world driving scenes) [8].

While I2I models have demonstrated remarkable performance in generating images that *appear* realistic to human observers [9], thus narrowing the inconsistent behavior between simulated and real-world data, they also bear notorious limitations in terms of the quality of their generated outputs, including issues like feature blending, color bleeding, or object omissions [2]. For this reason, it is important to ground the progress done in sim2real mitigation for ADS testing using established metrics that can measure the extent to which the gap between simulated and real-world data is reduced thanks to I2I models.

Unfortunately, there is still no consensus on which evaluation metrics are the most effective for this task. While studies [9], [10] have attempted to benchmark generative adversarial networks, including I2I models, additional research is required to determine whether these metrics are correlated with a reduction in sim2real gap of ADS that rely on neural-generated data as inputs.

The objective of this paper is to systematically investigate the benefits and limitations of employing I2I models for mitigating the sim2real gap in the context of ADS testing. Our

goal is to assess whether I2I models are a viable and effective means of bridging this gap. Our study has three primary objectives. First, we aim to assess whether the effectiveness of I2I models generalizes across various ADS tasks, or if certain I2I solutions are better suited for specific tasks. Second, we aim to assess the relationship between the existing metrics used to evaluate the quality of I2I model outputs and the performance, or lack thereof, of the ADS systems being tested. Finally, our third objective is to explore whether domain-specific fine-tuning of the existing metrics can improve their ability to correlate with the ADS behavior.

To address these objectives, we conducted a comprehensive evaluation involving two well-known I2I models, pix2pix and CycleGAN, in which we investigated their capabilities in mitigating the gap between simulation and reality for two critical ADS tasks, vehicle detection, and lane keeping. Additionally, we employed 13 existing metrics from the literature, both at the distribution level and at the single-image level, and assessed their correlation with the ADS behavior in terms of prediction error, confidence measured as a complement of DNN uncertainty, and attention error based on the heatmaps from the explainable artificial intelligence domain. Finally, we conducted task-specific fine-tuning of perception metrics, to explore whether this adaptation could improve its effectiveness in sim2real gap mitigation.

Our results revealed that the effectiveness of I2I models varies across different ADS tasks, and existing evaluation metrics do not consistently align with the ADS behavior. However, through task-specific fine-tuning of perception metrics, we achieved a more robust correlation than all other metrics of our study, for both I2I models and ADS tasks. This suggests that this task-specific perception metric can serve as a reliable indicator for assessing the reduction of the sim2real gap in model-level ADS testing.

Our paper makes the following contributions:

Evaluation An empirical validation of 13 existing metrics for I2I models in addressing the sim2real gap in ADS testing.

Perception Metric A task-specific perception metric to evaluate I2I models for sim2real scenarios in ADS testing.

Dataset An extension of an existing dataset for lane-keeping ADS simulator with precise segmentation maps.

II. IMAGE-TO-IMAGE TRANSLATION MODELS

Generative models represent a class of machine learning models designed to produce novel data resembling a provided dataset. This is achieved by capturing the underlying patterns and structures within the data, enabling the generation of novel data samples. Among these methods, Generative Adversarial Networks (GANs) [11] emerged as the most popular solution. GANs consist of two neural networks, namely the Generator and the Discriminator, that engage in a competitive learning process known as adversarial learning. The Generator's objective is to create synthetic data that is indistinguishable from real samples, while the Discriminator learns to accurately differentiate between them.

In the context of autonomous driving, GANs have been applied to various tasks such as input pre-processing, including raindrop removal [12], de-blurring [13], and de-hazing [14]. They also found applications in sim2real I2I translation tasks [15], [16], [17], [18], in which one significant challenge is ensuring that the generated images not only look realistic but also preserve the essential content and details of the input, i.e., their *semantics*. Since adversarial training may not guarantee semantics preservation, researchers have proposed GAN variations that introduce constraints on the Generator's output. Instances of these solutions are pix2pix [19], UNIT [20], DiscoGAN [21], and CycleGAN [22].

We focus our investigation on two popular I2I models, namely pix2pix [19] and CycleGAN [22], for their significant impact (the original papers have more than 21k citations each as of 16 January 2024), including software engineering applications for ADS testing [15], [23], [24].

pix2pix. pix2pix is an I2I model based on conditional Generative Adversarial Networks (cGANs) [25]. Unlike traditional GANs, the generator is provided with conditional information, such as an input image or label, in addition to random noise.

In the case of pix2pix, the conditional information is an input image from the source domain (e.g., real-world driving images) that needs to be translated into the target domain (e.g., synthetic driving images), which forces the generator to produce data that aligns with the input. The discriminator, as in a GAN network, is tasked with distinguishing between a real target image and a generated one. However, in pix2pix, the adversarial loss drives the Generator towards producing realistic images, whereas the L1 loss (Mean Absolute Error) ensures pixel-level accuracy, enforcing the creation of contextually accurate generated images.

CycleGAN. CycleGAN targets unsupervised I2I translation, i.e., it does not require paired training data from the source and target domains. CycleGAN adopts four neural networks, two generators (G_1 and G_2) and two discriminators (D_1 and D_2), which are trained simultaneously. The generator G_1 performs the sim2real translation from the source domain (e.g., simulated driving images) to the target domain (e.g., real-world driving images). Conversely, the generator G_2 performs the inverse real2sim transformation. The discriminator D_1 is trained to differentiate between real images from the target domain and images generated by G_1 , whereas the discriminator D_2 fulfills a similar role for the inverse mapping.

While the generators G_1 and G_2 employ adversarial training to generate images that closely resemble images from the target domain, the preservation of the original image's content (e.g., vehicles, pedestrians, or traffic signs occurring in a real-life driving scenario) is not ensured. Thus, to ensure that the translation is both realistic and semantics-preserving, CycleGAN uses a cycle consistency loss to quantify the disparity between the source image and the image obtained by cycling through the translation process. In other words, when an image is translated from the source domain to the target domain and then back to the source domain, the resulting image should closely resemble the original source image.

III. IMAGE-TO-IMAGE MODELS EVALUATION METRICS

Researchers have identified issues within GANs, including I2I models, that could potentially affect systems reliant on the use of their generated images as input. These include visual anomalies such as color bleeding, feature blending, omission, hallucination, erroneous object textures, and pixel patterns which can introduce inaccuracies and inconsistencies, disrupting ADS related to object recognition, segmentation, and scene understanding tasks [26].

The objective of this study is to explore the advantages and drawbacks of employing I2I models for sim2real gap mitigation in the context of ADS testing. Our specific goals are twofold. First, we aim to assess whether the effectiveness of I2I models can be transferred to various ADS tasks or if certain I2I solutions are better suited for specific ones. Second, we aim to assess whether the existing metrics used to evaluate the quality of I2I model outputs are correlated with the performance of the ADS being tested and whether improvements to the existing metrics can be made.

In this study, we evaluate the image quality assessment metrics provided by Borji [9] and by Pang et al. [10]. The selected metrics pertain to two categories, namely distribution-level metrics and single-image metrics.

A. Distribution-level Metrics

Distribution-level metrics are designed to evaluate the overall performance of an I2I translation model by considering the entire set of generated images as a collective output. These metrics provide insights into the global characteristics of the generated distribution, enabling them to work even if a mapping between the source and target domain is not available.

Inception Score [27] (IS). This metric assesses the quality and diversity of image distributions as follows. IS employs the pre-trained Convolutional Neural Network (CNN) model, Inception [28], as a feature extractor to compute classification confidence (which is used as a proxy for quality) and entropy of the predicted class probabilities (which is considered as a measure of diversity) on all generated images. Therefore, IS consists of an aggregate value computed as the exponential of the entropy of the average class probabilities. A high IS score indicates high-quality and more diverse images.

Fréchet Inception Distance [29] (FID). This metric quantifies the distance between the distributions of features extracted from real and generated images. Similarly to IS, the FID metric relies on a pre-trained Inception model as a feature extractor. The extracted features of both real and generated images are then modeled as a multivariate Gaussian Distribution and compared using the Fréchet distance [30]. A lower distance should indicate high similarity between the real and generated distributions, indicating a well-performing generative model.

Kernel Inception Distance [31] (KID). This metric quantifies the similarity between the distributions of feature representations by modeling them as kernel mean embeddings. The Gram matrix [32] is computed for both the real and generated images to capture relationships between their features. The square root of Maximum Mean Discrepancy (MMD) [33] is then used to

quantify the difference between the distributions based on the Gram matrices. A lower MMD score should indicate a better alignment between the two image distributions.

B. Single-Image Metrics

Single-image metrics focus on evaluating the quality of individual images generated by a model. These metrics assess how well each output image aligns with a ground truth or reference image. Due to the direct comparison between two images, these techniques require the input and target domain to have an explicit mapping. We categorize single-image metrics in *content-based metrics* and *perception-based metrics*.

1) *Content-based Single-Image Metrics:* These metrics only use the input image pair attributes to evaluate image differences. We consider seven content-based metrics.

Structural Similarity Index (SSIM) [34]. It evaluates the quality of a generated image by comparing its structural information, such as edges, textures, and patterns, to a reference image. The SSIM output is obtained by a combination of luminance (the difference in average intensity values), contrast (the distance between the image contrast distributions), and structure (the normalized covariance between the pixel intensities of the two images). A higher SSIM value indicates greater similarity between the two images in terms of their structural information, suggesting higher image quality.

Peak signal-to-noise ratio (PSNR) [27]. It measures the difference between an image pair by calculating the ratio of the maximum possible pixel value (peak signal) to the Mean Square Error (MSE) between the reference image and the reconstructed/generated image pixel values (noise). A high PSNR value should indicate a low average error between the two image domains thanks to a well-performing model.

Mean Squared Error (MSE). It quantifies the average squared difference between the pixel values of two images. Lower MSE values indicate higher similarity at the pixel level between the generated and the reference image [34].

Cosine Similarity (CS) [35]. It is a metric used to quantify the similarity between two feature representations. It calculates the cosine similarity of the feature vectors extracted using the same pre-trained classification models as in CPL. A higher cosine similarity value suggests that the images are more similar in terms of the features.

Texture Similarity Index (TSI). It is used to measure the texture properties of a generated and real image pair. For each of the two inputs, multiple Gray-Level Co-occurrence Matrices (GLCMs) [36] are computed and used to capture spatial relationships between pixel values. Using the output of these matrices it is possible to analyze how the textures differ in terms of properties like contrast, dissimilarity, homogeneity, energy, and correlation. The metric output is an aggregate of the absolute difference in texture properties between the two images. A lower texture similarity score suggests two images with close texture properties.

Wasserstein Score (WD) [37]. Also known as Earth Mover's Distance (EMD) [38], it assesses the similarity between two images by considering the cost of transforming one image

into the other. It quantifies the minimum amount of work (or distance) required to redistribute the mass from the pixels in the generated image to match the reference image. A lower Wasserstein Score indicates greater similarity.

KL Divergence (KL) [39]. It is a measure of the difference between two probability distributions. In the context of images, it quantifies the difference in pixel value distribution between a generated image and a reference image. A lower KL Divergence indicates greater similarity in pixel value distribution.

Histogram Intersection (HistI) [40]. It measures the similarity between two images by computing the intersection of their histograms, i.e., the distribution of pixel values. A higher intersection value suggests greater similarity.

2) *Perception-based Single-Image Metrics*: These metrics evaluate the differences between a real and a generated image by observing high-level features computed leveraging a pre-trained model for feature extraction, such as ResNet50 [41] or VGG [42]. We consider three perception-based metrics.

Classifier Perceptual Loss (CPL) [43]. It measures the difference between the feature maps created by the convolutional layers of a pre-trained classification model. Feature maps capture high-level representations of the input images, such as textures, objects, and shapes. By comparing the feature maps of a generated image to those of a reference image, perceptual loss aims to measure if the generated image not only matches pixel values but also retains analogous perceptual features. A lower perceptual loss value indicates that two images share more perceptual characteristics, suggesting high similarity.

Semantic Segmentation Score (SSS). It measures the semantic details and the structural difference between two images by applying a pre-trained segmentation architecture such as FCN-8 [44] on the generated image, and by comparing its output with the ground truth segmentation label of the source image. A high error between the semantic representation of the original and the predicted image should indicate a lack of fidelity in terms of image content.

IV. EMPIRICAL STUDY

The goal of our empirical study is to evaluate different I2I models for sim2real gap mitigation in the context of ADS testing. We consider two popular, yet different tasks related to autonomous driving, namely vehicle detection and lane keeping. These components are deemed essential by the U.S. Department of Transportation, National Highway Traffic Safety Administration (NHTSA), but they require rigorous testing to ensure that the vehicle can effectively stay in its lane and recognize pertinent objects before deployment [45].

A. Research Questions

We consider the following research questions.

RQ₁ (Transferability): *Does the effectiveness of I2I models for sim2real mitigation vary for different tasks?*

The first research question evaluates the effectiveness of I2I models for sim2real gap mitigation in the context of the two considered ADS testing tasks.

RQ₂ (Correlation): *How do existing I2I evaluation metrics correlate with the associated ADS behavior?*

In the second research question, we aim to assess whether our evaluated metrics are informative with respect to the (mis)behavior of ADS that use these inputs for sim2real gap mitigation.

RQ₃ (Fine-tuning): *Does fine-tuning of I2I perception-based metrics improve the sim2real mitigation task?*

Finally, in the third research question, we further fine-tune the perception metrics on each individual task to assess whether improvements can be achieved beyond the original metric propositions.

B. Datasets

1) *Vehicle Detection Dataset*: To perform translation quality evaluations for sim2real vehicle detection tasks, we adopt the KITTI [46] and vKITTI [47] datasets. We use vKITTI (sim) and KITTI (real) as source and target datasets for I2I models (Figure 1(a-b) shows an example).

More in detail, the KITTI dataset (Karlsruhe Institute of Technology and Toyota Technological Institute) is a widely used benchmark for autonomous driving perception research [48], [49], [50]. It provides a large collection of real-world data (e.g., GPS, image, and LIDAR sensors) captured from a moving vehicle in multiple road environments. The ground truth is represented by bounding boxes (i.e., rectangular regions) that specify the position and extent of the objects of interest. Additionally, semantic segmentation maps assign specific object class labels to individual pixels, allowing for pixel-level identification of different object categories (e.g., vehicles, road pavement). The vKITTI dataset (Virtual KITTI) is a virtual replica of a subset of KITTI, rendered using the Unity computer graphics engine [51]. The dataset was obtained by performing a semi-supervised real-to-virtual world cloning method where object positions and orientations in the virtual world were calculated based on their relationships with the camera and road positions in KITTI. Differently, environmental objects such as trees, buildings, and secondary roads were manually placed in the environment [47]. The final dataset includes 1,064 sim2real image pairs and labels across KITTI and vKITTI.

2) *Lane Keeping Dataset*: The second dataset involves imitation-learning-based end-to-end lane-keeping across virtual and real-world environments. We have considered the datasets from recent work on sim2real gap estimation for ADS driving [15], in which the authors provide a dataset of images captured with a physical small-scale autonomous vehicle performing lane-keeping on a real-world indoor track and the associated driving simulator (Figure 1(c-d) shows an example). Specifically, the data is collected using the DonkeyCar [52], a framework that includes a 1/16th scale vehicle and a digital twin simulation platform (Figure 1(c-d) shows an example). In total, the dataset includes 14k simulated driving frames, of which 5k are unlabeled and 9k labeled, and 21k driving frames, of which 6k unlabeled and 15k labeled.



(a) Vehicle Detection Sim. (b) Vehicle Detection Real. (c) Lane Keeping Sim. (d) Lane Keeping Real.

Fig. 1: An example of dataset sim2real pairs for the Vehicle Detection (a-b) and Lane Keeping (c-d) datasets.

C. ADS Under Test

1) Vehicle Detection: Concerning the vehicle detection task, we adopted the YOLOv3 [53] model (You Only Look Once v.3), an object detection and classification CNN architecture that has been adopted in the autonomous driving domain due to its accuracy and performance on constrained hardware [54], [55]. We chose version 3 for its maturity, its availability of robust pre-trained model weights [56], [57], and its higher inference speed compared to more recent variations [58]. In this study, we use a model pre-trained on the Microsoft Common Objects in Context dataset (COCO) [59], an image dataset with semantic labels for 50 object classes and 123k samples, of which 13k pertain to vehicles. The model outputs for each detected object are the object class, the confidence of said classification, and the bounding box containing the object. For this study, we only consider object predictions pertaining to vehicles due to a marginal presence of pedestrians, bicycles, and other classes relevant to ADS tasks in the subset of (v)KITTI. In the rest of the paper, we shall use the terms object/vehicle detection dataset interchangeably.

2) Lane Keeping: DAVE-2 is a CNN model for lane-keeping tasks based on imitation learning [60]. Given an input image representing a road scene, DAVE-2 processes it through three convolutional layers for feature extraction followed by five fully connected layers for steering angle regression. By providing recorded input images associated with steering commands, this model is capable of learning how to keep a vehicle on the road. Due to its performance despite its simple architecture, this model has been used for a large variety of ADS testing studies [15], [23], [61]. More importantly, DAVE-2 exhibits realistic behaviors in simulated environments, with performance degradation [62] that closely mimics reality when the generated images exhibit distortions in the lanes [63].

D. ADS behavior Metrics

To evaluate the behavior of both ADS, we conduct a comparison between ground truth annotations of real-world images and the ADS outputs generated using simulated, I2I translated, and corresponding real images. This evaluation is based on three ADS behavioral metrics [64]: prediction error, confidence, and attention error.

Prediction Error. The first behavioral metric is prediction error, i.e., the discrepancy between the predicted output of the ADS and the ground truth value. Intuitively, when the I2I model generates poor translations, it leads to high ADS prediction errors, signifying ineffective sim2real gap mitigation,

while high-quality translations result in low prediction errors, indicating successful mitigation.

For YOLOv3, we use the Intersection over Union (IoU) [65] metric to assess the alignment between predicted bounding boxes and the ground truth bounding boxes. Any significant misalignment leads to false positives and false negatives, whose severity is further assessed by considering their respective areas. For example, a false negative that involves a vehicle in the foreground is deemed more critical than one in the background. For instances in which the bounding boxes are correctly matched, i.e., a successful object detection, we also measure the area difference between the two boxes to quantify the degree of bounding box misalignment. This methodology enables us to assess the accuracy of vehicle detection as well as to evaluate the extent of the bounding boxes' error prediction. For DAVE-2, we compute the MSE between the predicted and ground truth steering angle.

Confidence. The second behavioral metric consists of a measure of confidence associated with the ADS prediction. We follow the intuition that this metric can help evaluate the quality of I2I translated inputs as high confidence may suggest a high degree of faith in the accuracy of the transformation, whereas low confidence may indicate a higher likelihood of discrepancies between real and translated images. In YOLOv3, the confidence of class predictions is readily available as one of the outputs. To assess the overall confidence of the model for one input, we simply aggregate the confidence scores for all predicted vehicles, resulting in a single averaged confidence value for the vehicle class. In contrast, the standard configuration of DAVE-2 provides only inference outputs, i.e., steering angles. Thus, following existing guidelines [66], [67], we use the predictive variance of dropout-based DNNs called Monte Carlo dropout, or MC-Dropout (MCD) [68]. MCD leverages dropout layers, where random neurons are deactivated during inference, to inject randomness into the ADS predictions. By carrying out multiple iterations with dropout applied, MCD generates a range of predictions for the same input. The variance of the observed probability distribution quantifies the uncertainty (i.e., the confidence error).

Attention Error. The last ADS behavioral metric relies on attention maps generated through explainable artificial intelligence techniques (XAI). Attention maps are post-training methods that highlight the specific input pixels that exert the most significant influence on the output predictions [69]. In our study, we employ the GradCam algorithm [70], a widely used technique based on gradient back-propagation which enables us to gain insights into the image areas that the ADS considers

TABLE I: Datasets Usage (✓ used; ✗ not used).

	Data split (set)	Used for	Size [sim/real]	ADS labels	Paired	Seg. maps
Vehicle Detection	Implementation	I2I/Seg	[1,064/1,064]	✓	✓	✓
	Evaluation	RQs	[1,064/1,064]	✓	✓	✓
Lane Keeping	Train	ADS	[-/7905]	✓	✗	✗
	Implementation	I2I/Seg	[5,361/5,361]	✗	✗	✓
	Evaluation	RQs	[197/197]	✓	✓	✓

relevant during its execution. We utilize the attention map as a quality indicator to determine if the ADS is directing its attention to image regions pertinent to its intended task. When the attention map indicates ADS focus on areas different from those obtained on real-world data, it could indicate that the translation process has impacted the ADS reliability with respect to the unaltered real-world data.

For each input image, we use as ground truth the corresponding real-world semantic segmentation map, which provides a pixel-level identification of different objects and regions within the image. We use it to automatically detect the areas of interest (i.e., the vehicles for vehicle detection and road lanes for lane keeping) and compare them with the scores of the attention map generated by the I2I model for that image. Specifically, we apply a binarization process to both maps. For attention maps, we employ a threshold that highlights image areas of significant importance to the model, retaining those with an intensity above 50%. In the semantic segmentation map, we assign a value of 1 to pixels that represent relevant objects, while setting all other pixels to 0. Subsequently, we compute the MSE between these two binary maps to quantify the model attention error.

E. Procedure

Dataset Pairing, Pre-processing, and Splitting. Our study requires splitting the simulated and real datasets into different independent sets, with different requirements in terms of pairing. Table I summarizes the dataset size and split used in the paper. For evaluating the I2I models in relation to the ADS behavior metrics, our study requires an *evaluation set* of *paired* simulated and real data. Without this consistency, assessing whether the model successfully maintains content while improving quality and style becomes challenging. A separate *implementation set*, for which no pairing is required, is used to train the I2I models as well as the segmentation model used in RQ₃. Finally, the lane-keeping case study also requires a *training set* for training DAVE-2.

The vehicle detection dataset already meets the pairing requirement as vKITTI is a synthetic counterpart of a subset from the KITTI dataset. Thus, we divided the paired data into two \approx 1,000-image sets for the *implementation set* and *evaluation set*, without further preprocessing. In contrast, for the lane-keeping dataset, we established a paired *evaluation set* of simulated and real labeled images from the original test set [15] by performing a manual alignment of 197

frames between simulated and real images.¹ The remaining (unpaired) data was divided into two groups: a *training set* of 7,905 labeled real images and an *implementation set* of 5,361 unlabeled simulated and real images.

To train pix2pix, due to its supervised nature, we need a substantial number of paired images. The vehicle detection dataset already includes the required semantic maps for pix2pix. In contrast, for the lane-keeping dataset, we generated segmentation maps by associating specific pixel values with semantic classes in each image (i.e., road and background), followed by manual refinement to ensure map accuracy.

Training I2I Models and Data Generation. To train both pix2pix and CycleGAN, we used the *implementation set*, as follows. pix2pix is trained using real images and their corresponding segmentation masks, whereas CycleGAN is trained using unpaired simulated and real images. We trained pix2pix for \approx 23 hours and CycleGAN for \approx 2.5 hours on a Google Cloud TPU v4 with 36GB of VRAM equivalent [71]. Both models are trained using the original code provided by the authors. During training, we saved the checkpoints (i.e., model weights) at the end of each epoch, which allowed us to consider models at various stages of maturity. After training, we selected three checkpoints for pix2pix and CycleGAN, approximately at 10-50-90% of the training time, to assess poorly/medium/largely trained models (Table II). Their respective generators were then used to produce a translated copy of the simulated data in the *evaluation set*.

TABLE II: Training statistics for the I2I models.

	I2I model	Training		Checkpoints		
		Epochs	Time (h)	Ckp. name	Epochs	Time (h)
Vehicle Detection	pix2pix	130	23.1	P1/P2/P3	14/70/120	2.4/12.4/21.3
	CycleGAN	38	2.63	C1/C2/C3	4/19/33	0.2/1.3/2.2
Lane Keeping	pix2pix	170	22.95	P1/P2/P3	20/80/150	2.7/10.8/20.2
	CycleGAN	40	2.42	C1/C2/C3	8/20/34	0.4/1.2/2

Training ADS. The upper bound number of epochs was set to 200, with a batch size of 128 images and a learning rate of 0.001. We used early stopping with a patience of 10 epochs on the validation set. The DNNs use the Adam optimizer [72] to minimize the MSE between the predicted steering angles and the ground truth value.

RQ₁ (Transferability). To answer RQ₁, for each ADS task, we execute the ADS models (YOLOv3 and DAVE-2) on the real-world *evaluation set* to collect the nominal prediction error, confidence values, and attention errors (Section IV-D). We then executed each ADS on the simulated *implementation set* and on the corresponding data generated by the six I2I translation models created for each task (three for pix2pix and three for CycleGAN) and collected the corresponding

¹While the size of the evaluation set of the lane-keeping dataset is lower than the one for the vehicle detection dataset, the number of images is sufficient to represent all the conditions of the real-world road track used in the previous paper [15], an 11 m long track, 52 cm wide. Clockwise, the track features three bends on the right and one on the left.

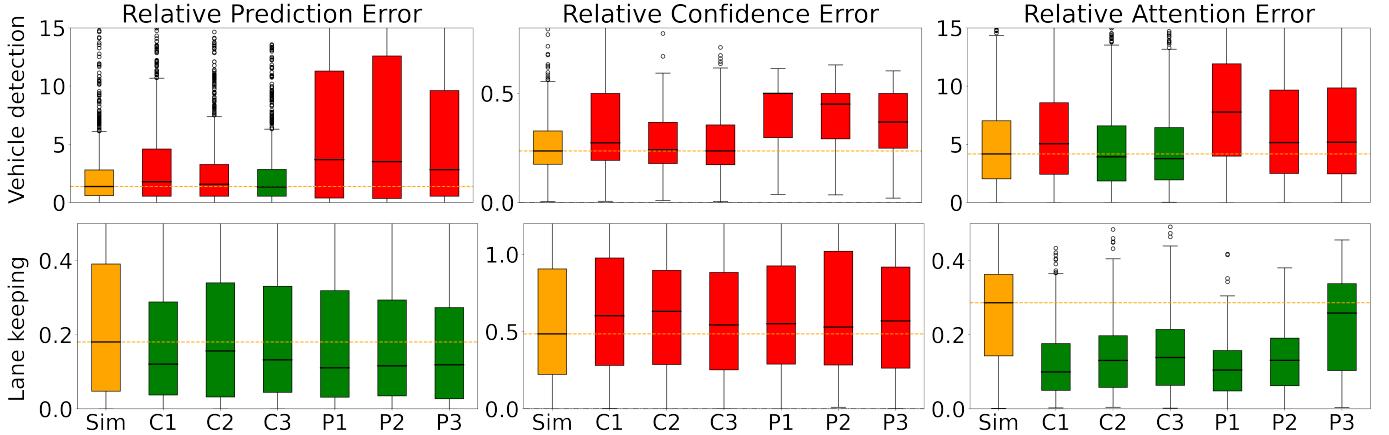


Fig. 2: Distributions of ADS behavior on different image domains. The black horizontal line represents the *median* of the distribution, whereas the color indicates whether the median is lower (green) or same/higher (red) as compared to the simulated domain (yellow); In each boxplot, distributions with lower values are preferable. The figure is best viewed in color.

ADS behavior metrics. Finally, we computed the relative error concerning the score of these metrics on real-world data, which provided a quantification of how the ADS behavior deviated from the expected behavior.

RQ₂ (Correlation). To address RQ₂, for both ADS tasks, we compute the metrics outlined in Section III-A on the I2I translated images from the evaluation set used for RQ₁. Then, we compute the correlation between these metrics and the relative errors of ADS behaviors using Pearson’s correlation coefficient [73]. For distribution-level metrics, we compute the correlation with the average of the relative ADS behavioral metrics. For single-image metrics, we compute their correlation with the relative ADS behavior for each input domain separately. To compute IS and FID, we utilize InceptionV3 [74], whereas for computing CPL, CS, and SSS, we used a VGG16 [42] model, both pre-trained on ImageNet [75].

RQ₃ (Fine-tuning). We develop a perception metric customized for our image domains through fine-tuning. Particularly, we leverage a SegFormer model [76] pre-trained on the Cityscapes dataset [77], a transformer-based segmentation model, renowned for image semantic segmentation tasks. Which is fine-tuned to the specific ADS tasks by training it only using segmentation masks that contain the classes relevant to each ADS task. For vehicle detection, we used the classes “vehicle”, “road”, “environment” and “sky”, whereas, for lane-keeping, we considered the classes “road” (including central and outer lanes) and “background”. Subsequently, we employ the trained SegFormer model to assess the semantic differences between the original image and the translated version, following a similar approach as for the SSS metric.

To offer varying levels of fine-tuning, we compute semantic difference values both (i) across all relevant classes (Targeted Semantic Segmentation TSS) and (ii) only for the most relevant class of each ADS task—i.e., “vehicle” for vehicle detection and “road” for lane keeping—(One Class - Targeted Semantic Segmentation OC-TSS). After computing the TSS

TABLE III: Distribution-level metrics relative error correlations.

	Vehicle Detection			Lane Keeping		
	Prediction	Confidence	Attention	Prediction	Confidence	Attention
IS	0.41	0.37	0.41	0.14	0.72	0.65
FID	0.24	0.21	0.31	0.64	0.86	0.78
KID	0.54	0.64	0.86	0.54	0.74	0.60

and OC-TSS scores, we computed the correlation between both fine-tuned image metric values and the ADS behavior metrics as done for RQ₂.

F. Results

RQ₁ (Transferability). Figure 2 (top) shows the results for the vehicle detection ADS, whereas Figure 2 (bottom) shows the results for the lane-keeping ADS. For both ADS tasks, each row displays several boxplots representing the ADS behavior deviation from the real counterpart (prediction error, confidence, and attention error) on simulated data (Sim) and I2I-generated data for the I2I models under test at various epochs (C1, C2, C3, P1, P2, P3, see Table II).

Considering the relative prediction and attention errors, C3 has a good performance in mitigating the sim2real gap in the vehicle detection task, whereas, for the lane-keeping task, our results show both I2I models mitigate the sim2real gap. Our results reveal that the relative confidence error may not be useful for sim2real gap mitigation assessment, as none of the models of either domain show an improvement over the simulated inputs. We assess the statistical significance of the differences between real-world and simulator/I2I errors using a paired t-test [78] (with $\alpha = 0.05$), the magnitude of the differences using the Cohen’s d effect size [79], and the statistical power with a Monte Carlo power analysis [80] with 80% power target as our data is not normally distributed.

TABLE IV: Single-image metrics correlations. **Bold values** indicate statistical significance ($p\text{-value} < 0.05$), **red values** indicate wrong correlation direction, and **grey values** indicate correlations lower than 0.10. \downarrow/\uparrow = lower/higher values are preferable.

	Relative Prediction Error						Relative Confidence Error						Relative Attention Error						
	C1	C2	C3	P1	P2	P3	C1	C2	C3	P1	P2	P3	C1	C2	C3	P1	P2	P3	
RQ₂																			
VEHICLE DETECTION	SSIM \uparrow	-0.03	0.06	-0.02	0.18	0.21	0.26	0.16	0.11	0.20	0.06	0.05	0.08	-0.09	-0.04	-0.02	-0.09	-0.14	-0.26
	PSNR \uparrow	0.08	0.07	-0.09	0.30	0.32	0.35	0.18	0.04	0.14	-0.01	0.01	0.12	-0.11	-0.01	-0.01	-0.01	-0.08	-0.16
	MSE \downarrow	-0.08	-0.08	0.09	-0.30	-0.31	-0.35	-0.18	-0.05	-0.14	0.01	-0.01	-0.12	0.11	0.02	0.00	0.00	0.08	0.16
	CS \uparrow	-0.03	0.04	-0.06	-0.21	-0.14	0.18	0.18	0.13	0.04	-0.01	0.09	0.20	0.10	0.03	0.01	0.32	0.20	-0.11
	TSI \downarrow	0.07	0.00	-0.13	-0.08	-0.13	-0.16	0.05	0.01	-0.12	0.04	-0.05	-0.01	-0.20	0.15	0.13	-0.09	0.15	0.37
	WD \downarrow	-0.08	0.00	0.04	0.06	0.09	0.09	0.05	0.07	0.31	0.17	0.11	0.15	-0.10	0.00	-0.06	-0.30	-0.28	-0.35
	KL \downarrow	0.03	-0.13	0.14	-0.11	0.04	0.12	0.07	-0.11	0.13	0.04	0.18	0.14	-0.08	0.15	-0.11	0.18	-0.08	-0.18
	Histi \uparrow	0.09	0.05	-0.06	0.23	0.17	0.31	0.05	0.10	-0.06	-0.00	-0.09	0.06	-0.01	-0.12	0.07	0.02	0.03	-0.27
	CPL \downarrow	0.00	0.01	0.02	0.29	0.20	0.16	0.23	0.19	0.11	0.00	0.15	0.08	-0.10	-0.01	-0.01	0.17	0.26	0.36
	SSS \downarrow	-0.02	-0.01	0.01	0.23	0.19	0.10	0.21	0.17	0.13	0.01	0.18	0.09	-0.12	-0.04	-0.03	0.21	0.23	0.34
RQ₃																			
LANE KEEPING	TSS \downarrow	0.05	-0.08	-0.09	0.26	0.14	0.01	-0.05	-0.02	-0.11	-0.04	0.05	-0.03	0.04	0.19	0.13	0.11	0.21	0.36
	OC-TSS \downarrow	0.33	0.21	0.15	0.57	0.55	0.49	0.13	0.19	0.21	0.08	0.10	0.12	-0.10	-0.08	-0.09	-0.24	-0.19	-0.24
	RQ₂																		
	SSIM \uparrow	-0.31	-0.28	-0.39	-0.09	-0.03	-0.15	-0.14	-0.13	-0.05	-0.12	-0.07	-0.13	-0.34	-0.37	-0.20	-0.03	0.05	0.35
	PSNR \uparrow	-0.33	-0.34	-0.45	-0.12	-0.08	-0.12	-0.11	-0.02	-0.08	-0.13	-0.01	-0.11	-0.38	-0.36	-0.31	-0.02	-0.10	0.18
	MSE \downarrow	0.34	0.34	0.45	0.12	0.07	0.12	0.11	0.02	0.08	0.13	0.01	0.11	0.38	0.36	0.31	0.01	0.10	-0.17
	CS \uparrow	0.03	0.00	-0.07	-0.06	-0.14	0.02	-0.04	-0.10	-0.05	-0.03	-0.13	0.02	-0.14	-0.01	-0.06	-0.11	0.14	-0.00
	TSI \downarrow	-0.15	-0.12	-0.06	-0.26	-0.18	0.00	0.10	0.11	0.20	-0.10	-0.14	-0.07	-0.04	-0.07	-0.18	-0.18	-0.16	
	WD \downarrow	0.07	0.02	-0.15	-0.07	-0.07	-0.16	0.10	0.07	0.07	0.13	0.10	0.07	-0.05	0.00	-0.08	-0.19	0.03	0.22
	KL \downarrow	0.25	0.10	0.10	-0.05	0.00	-0.05	0.21	0.10	0.07	0.11	0.18	0.19	0.27	0.29	0.37	-0.01	-0.09	-0.12
	Histi \uparrow	-0.15	0.01	-0.24	0.21	0.03	-0.27	0.11	0.08	0.13	0.11	0.20	-0.01	-0.13	0.10	-0.17	0.05	-0.37	0.03
	CPL \downarrow	0.15	0.28	0.30	0.22	0.17	0.02	0.18	0.25	0.12	0.15	0.19	0.30	-0.05	-0.13	-0.13	0.01	0.14	0.16
	SSS \downarrow	0.21	0.18	0.25	0.11	-0.10	-0.05	0.23	0.30	0.12	0.21	0.19	0.26	-0.02	-0.05	-0.11	-0.02	0.26	0.07
RQ₃																			
LANE KEEPING	TSS \downarrow	0.15	0.16	0.16	0.17	0.13	0.20	0.15	0.16	0.16	0.17	0.13	0.20	0.07	-0.01	-0.17	0.08	-0.10	-0.48
	OC-TSS \downarrow	0.39	0.35	0.44	0.42	0.31	0.38	0.14	0.15	0.15	0.18	0.13	0.19	0.06	-0.04	-0.16	0.10	-0.08	-0.45

In the vehicle detection task, only C3 has a distribution of prediction errors for which it was the same for real-world and for I2I-generated images (i.e., $p\text{-value} \geq 0.05$, and high statistical power). This indicates that a high-quality CycleGAN architecture can generate sim2real gap mitigating inputs for which the ADS outputs predictions similar to those on real-world data. Differently, none of the domains showed statistically significant differences for confidence and attention errors ($p\text{-value} < 0.05$). These findings indicate that such ADS behavioral metrics are less adequate to analyze the ADS transferability across virtual and real-world environments at the model level. In the lane-keeping task, C2, C3, and P2 have a distribution of prediction and attention errors that were the same for the real world and for I2I-generated images (i.e., $p\text{-value} \geq 0.05$, and high statistical power). Regarding confidence, our analysis does not identify any distribution as statistically close.

RQ₁ (Transferability) The effectiveness of I2I models for mitigating the sim2real gap varies across different tasks. For lane keeping, all I2I models mitigate the sim2real gap in terms of prediction and attention errors; for vehicle detection, only a high-quality CycleGAN achieves this outcome.

RQ₂ (Correlation). Concerning distribution-level metrics, Table III presents the Pearson’s correlation coefficients between the distribution-level I2I models evaluation metrics (first column) and the ADS evaluation metrics (remaining columns). Also in this case results are aggregated by ADS task. Bold values indicate statistical significance ($p\text{-value} < 0.05$).

Our results indicate good correlations between all distribution-level metrics and ADS metrics in both ADS tasks, except for IS and prediction error in the lane-keeping task. Specifically, in the vehicle detection task, the IS metric correlates with statistical significance with relative prediction error, and KID does so with attention error. In contrast, in the lane-keeping task, FID demonstrates a significant correlation with both Uncertainty and attention error, while KID does so with uncertainty.

Concerning image-level metrics, Table IV present the Pearson’s correlation coefficients between the single-image metrics (rows) and the ADS evaluation metrics (columns) on each generated domain (C1, C2, C3, P1, P2, P3) for both. Results are aggregated by ADS task (Vehicle detection and Lane keeping). Overall, our results show correlation ranges ∓ 0.45 ; for the sake of space, we limit our analysis to metrics having absolute correlation scores higher than 0.10, with statistical significance. For the vehicle detection task, CPL and SSS are the only metrics that exhibit a statistically significant and

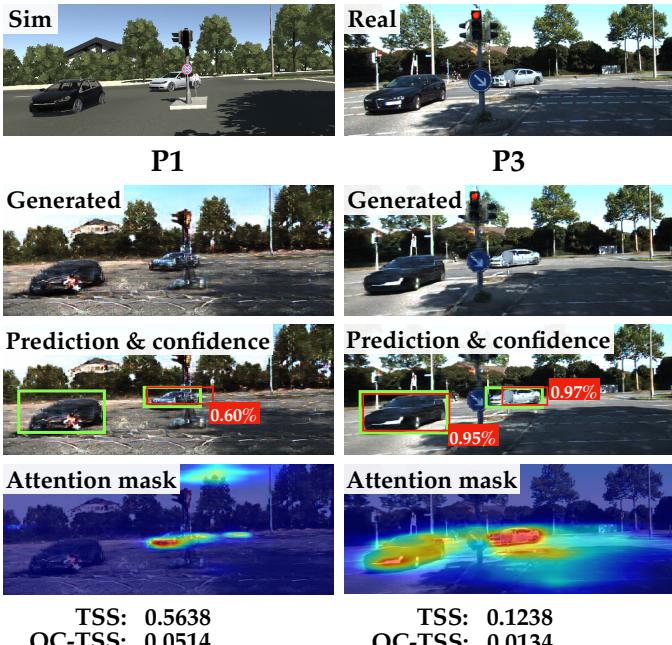


Fig. 3: Translation quality comparison: P1 (low-quality pix2pix) and P3 (high-quality pix2pix). Rows depict simulated/reference images, neural-generated images by P1 and P3, ground truth and vehicle detection ground truth and output (green and red boxes) with confidence (in percentage), ADS attention mask, and TSS/OC-TSS values—larger values of the metrics highlight the error between the semantic representation of the original and generated images, thus characterizing neural translation effectiveness.

consistent correlation, without directional disagreement, with the ADS relative prediction errors. This trend is observed exclusively in pix2pix-based models (P1, P2, and P3). On the other hand, in the lane-keeping task, SSIM, PSNR, MSE, and SSS exhibit this behavior for CycleGAN-based models, while CPL achieves it in both tasks. In terms of relative confidence error, WD, CPL, and SSS exhibit statistically significant correlations without directional discrepancies in the vehicle detection task, while, for the lane-keeping task, this is observed only with CPL and SSS. Lastly, concerning Attention error, SSIM, PSNR, and MSE show a correlation with some models in the vehicle detection task, whereas none of the metrics do so in the lane-keeping task.

RQ₂ (Correlation): All distribution-level metrics correlate well with at least one ADS behavior metric. The correlation does not transfer across ADS tasks. Single-image metrics, on the contrary, are highly inconsistent both across ADS tasks and I2I models, making them less suitable for sim2real ADS assessment.

RQ₃ (Fine-tuning). The bottom part of Table IV, shows, for each task, the Pearson’s correlation coefficients between the TSS and OC-TSS metrics (first column) and the ADS evaluation metrics (remaining columns).

The metric TSS, which utilizes all segmentation classes to calculate image difference, exhibits correlation scores with the ADS behavior in line with existing metrics (RQ₂) for both ADS tasks (values up to 0.36), but notably does not show directional correlation disagreement for prediction error and attention error in the vehicle detection task and for prediction error and confidence error in the lane-keeping task. Across ADS tasks, TSS correlates negatively with the attention error for P1 and P3 in the lane-keeping task, whereas it correlates positively in the vehicle detection task for all models.

Differently, OC-TSS is the only metric in our study that shows stable correlations across ADS tasks. Notably, it also achieves the highest correlation scores for P1, P2, and P3 (vehicle detection). It shows positive correlations with relative prediction errors in both ADS tasks, for both pix2pix and CycleGAN models.

Finally, it is worth noting that when it comes to the vehicle detection task, OC-TSS stands out as the only metric that consistently exhibits a statistically significant positive correlation with all evaluation metrics across various generative domains (specifically, pix2pix and CycleGAN). The only exception to this trend is the attention scores, which display a negative correlation, in contrast with the expected metric direction.

RQ₃ (Fine-tuning): In both evaluated task domains, TSS and OC-TSS provide valuable measures to evaluate the quality of I2I translation models, in terms of relative prediction and confidence error, making them suitable for sim2real ADS assessment.

Figure 3 shows some meaningful examples from our study. We present an example of low-quality and high-quality I2I translations produced by pix2pix models P1 and P3. For each model output (second row), we provide the predictions of the vehicle detection model, including confidence scores (third row), and the associated attention maps (fourth row). Below each column, we present the TSS and OC-TSS metrics that we have developed to address RQ₃. These metrics indicate a significant decline in quality, with a five-fold difference similar to the decrease in ADS performance, when comparing the visually inferior translation to the superior one. This drop in quality is evident in incomplete vehicle classification, increased false negatives, reduced confidence in predictions, and less focused attention.

G. Threats to Validity

1) *Internal validity:* In the absence of a consensus on appropriate metrics for evaluating I2I models, we encountered challenges in determining when to halt the training of our I2I translation architectures. Consequently, we extended the training of our generators until the generated images reached a visually high-quality standard. To ensure impartiality in our approach, we trained each architecture for an equal duration in both task domains and recorded model checkpoints. From these checkpoints, we arbitrarily selected three training epochs (10-50-90%) for reporting.

For both ADS task evaluations, the requirement was to have pairs of simulated and real images. Unfortunately, to our knowledge, the only existing lane-keeping dataset [81], paired images based on steering angles rather than sim-to-real image similarity. Consequently, we faced a shortage of suitable resources and had to resort to manual mapping of a simulated and real dataset [15]. These datasets depicted the same scenes, but they were not originally designed to provide a one-to-one correspondence between simulated and real images. This manual mapping process may have introduced minor content inconsistencies due to its non-standardized nature.

In the case of the DAVE-2 lane-keeping model used in one of our ADS domains, our results are contingent on the training quality of this model. We adhered to best training practices to ensure that the results reflected the behavior of a well-performing model. However, it is important to note that results may vary if the model is trained with different parameters.

2) External Validity: Regarding external validity, it is worth acknowledging that our study is limited by the number of ADS systems and I2I models considered, due to the experiment requirements (overall, we computed $340k$ scores = 6 I2I models \times 15 I2I metrics \times 3 ADS metrics \times $1,261$ images). This limitation can impact the generalizability of our findings.

3) Reproducibility: The entire pipeline used to obtain the results discussed in this empirical study, including I2I model training, segmentation models, metric calculations, and comparisons, is accessible and can be reproduced [82].

V. RELATED WORK

Existing research in the field of autonomous driving has emphasized the need for real-world testing of ADS. Simulation platforms, while valuable in the initial stages of testing, are characterized by limitations in faithfully replicating the complex characteristics of real-world environments [6], [83]. For a comprehensive exploration of the reality gap problem in ADS, we recommend referring to the existing literature [84].

Sim2real gap mitigation techniques are highly diverse, including digital twins [85], transfer learning [86], adversarial testing [87], multi-simulation environments [23], domain randomization [88], and hardware-in-the-loop testing [89].

In this paper, we focus on reality gap mitigation at an input level through the utilization of GANs for real-world like ADS inputs. In the context of model-level testing [24], [81], researchers have employed GAN-generated driving scenarios to evaluate the performance of these systems. DeepRoad [16] leverages UNIT [20] to generate highly realistic paired driving scenes for testing self-driving cars.

Concerning system-level testing, SilGAN [90] uses GANs to create driving maneuvers for software-in-the-loop testing. Kong et al. [91] developed realistic adversarial billboards embedded in real-world images to potentially confuse autonomous vehicles. DeepBillboard [87] employs both digital and physical adversarial perturbation techniques to influence offline steering. SurfelGAN [92] is a technique developed by Waymo to generate realistic sensor data for autonomous driving simulations without the need for manually creating

virtual environments and objects. A recent study [15] uses CycleGAN to generate real-world-like images from a driving simulator to train a telemetry predictor. Our findings reveal that the efficacy of CycleGAN can vary depending on the specific tasks, including factors related to ADS outputs.

To improve the capabilities of GANs and address image quality issues, numerous studies have examined different GAN evaluation metrics. However, there is no clear consensus on which metrics are most suitable for assessing GAN performance [9], [10], even less when considering the sim2real gap in ADS testing.

Some studies have empirically validated GANs but have primarily focused on analyzing the resulting distributions without specific task-oriented evaluations [93], or they have used general-purpose ImageNet models [94]. In our work, instead, we evaluate the available metrics in correlation with the ADS performance under test and propose a task-specific metric that captures the behavior differences caused by synthetic inputs generated by different GAN-based I2I translation models in the context of image-gap mitigation.

To the best of our knowledge, our study is the first that targets the reality gap mitigation for ADS inputs using generative I2I translation models and evaluates different metrics to quantify the gap between virtual and real imagery data for ADS testing.

VI. CONCLUSIONS AND FUTURE WORK

This work addresses the sim2real gap for vision-based ADS by employing data-driven neural I2I translation models to transform simulated driving scene images into real-world driving counterparts. I2I models can generate realistic images but have limitations, making it essential to have reliable metrics that can assess whether the inputs generated by such techniques are sim2real gap mitigating.

To this aim, this study contributes by empirically evaluating 13 existing metrics and two I2I models in addressing the sim-to-real gap in ADS testing. Our study highlights the limited transferability of I2I models across ADS domains and it reveals that these metrics lack consistency across tasks. This issue is addressed by introducing a task-specific perception metric, which is shown to have higher transferability.

Concerning future work, we plan to employ the best metrics from our study to develop data-driven techniques for sim2real-mitigating simulation-based testing of ADS. Additionally, these metrics can be utilized in combination with input validators [67], [69], [95], [96], [97] to filter out low-quality real-world data that are not expected to be supported by the ADS. Recent propositions showed the ability of diffusion models and neural radiance fields to generate high-quality images, which we plan to study as well in our future work. Additionally, we plan to expand our study to other I2I models and ADS tasks, including evaluations at a system level employing driving simulation platforms.

ACKNOWLEDGEMENTS

This research was funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy.

REFERENCES

- [1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.
- [2] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, “End-to-end autonomous driving: Challenges and frontiers,” 2023.
- [3] V. G. Cerf, “A comprehensive self-driving car test,” *Commun. ACM*, vol. 61, no. 2, pp. 7–7, Jan. 2018.
- [4] Z. Zhong, Y. Tang, Y. Zhou, V. de Oliveira Neves, Y. Liu, and B. Ray, “A survey on scenario-based testing for automated driving systems in high-fidelity simulation,” 2021.
- [5] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, “Testing Machine Learning based Systems: A Systematic Mapping,” *Empirical Software Engineering*, 2020.
- [6] A. Afzal, D. S. Katz, C. Le Goues, and C. S. Timperley, “Simulation for Robotics Test Automation: Developer Perspectives,” in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2021, pp. 263–274.
- [7] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y.-F. Li, L. Ma, Y. Xue, and Y. Liu, “A survey on automated driving system testing: Landscapes and trends,” 2022.
- [8] S. W. Kim, J. Philion, A. Torralba, and S. Fidler, “DriveGAN: Towards a Controllable High-Quality Neural Simulation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021.
- [9] A. Borji, “Pros and Cons of GAN Evaluation Measures,” *CoRR*, vol. abs/1802.03446, 2018.
- [10] Y. Pang, J. Lin, T. Qin, and Z. Chen, “Image-to-image translation: Methods and applications,” *IEEE Transactions on Multimedia*, vol. 24, pp. 3859–3881, 2022.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.
- [12] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, “Attentive generative adversarial network for raindrop removal from a single image,” *CoRR*, vol. abs/1711.10098, 2017.
- [13] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, “Deblurgan: Blind motion deblurring using conditional adversarial networks,” *CoRR*, vol. abs/1711.07064, 2017.
- [14] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, “An all-in-one network for dehazing and beyond,” *CoRR*, vol. abs/1707.06543, 2017.
- [15] A. Stocco, B. Pulfer, and P. Tonella, “Mind the Gap! A Study on the Transferability of Virtual Versus Physical-World Testing of Autonomous Driving Systems,” *IEEE Transactions on Software Engineering*, vol. 49, no. 04, pp. 1928–1940, apr 2023.
- [16] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, “Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems,” in *Proceedings of ASE ’18*, ser. ASE 2018. New York, NY, USA: ACM, 2018, pp. 132–142. [Online]. Available: <http://doi.acm.org/10.1145/3238147.3238187>
- [17] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” *CoRR*, vol. abs/1612.07828, 2016. [Online]. Available: <http://arxiv.org/abs/1612.07828>
- [18] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, “Cycada: Cycle-consistent adversarial domain adaptation,” *CoRR*, vol. abs/1711.03213, 2017. [Online]. Available: <http://arxiv.org/abs/1711.03213>
- [19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *CVPR*, 2017.
- [20] M. Liu, T. M. Breuel, and J. Kautz, “Unsupervised Image-to-Image Translation Networks,” *CoRR*, vol. abs/1703.00848, 2017. [Online]. Available: <http://arxiv.org/abs/1703.00848>
- [21] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” *CoRR*, vol. abs/1703.05192, 2017. [Online]. Available: <http://arxiv.org/abs/1703.05192>
- [22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
- [23] M. Biagiola, A. Stocco, V. Riccio, and P. Tonella, “Two is better than one: Digital siblings to improve autonomous driving testing,” 2023.
- [24] A. Stocco, B. Pulfer, and P. Tonella, “Model vs system level testing of autonomous driving systems: A replication and extension study,” *Empirical Softw. Engng.*, vol. 28, no. 3, may 2023.
- [25] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *CoRR*, vol. abs/1411.1784, 2014.
- [26] A. Borji, “Qualitative failures of image generation models and their application in detecting deepfakes,” 2023.
- [27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” 2016.
- [28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2014.
- [29] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” 2018.
- [30] M. Fréchet, “Sur la distance de deux lois de probabilité,” *Annales de l'ISUP*, vol. VI, no. 3, pp. 183–198, 1957.
- [31] J. Yu, X. Xu, F. Gao, S. Shi, M. Wang, D. Tao, and Q. Huang, “Towards realistic face photo-sketch synthesis via composition-aided gans,” 2020.
- [32] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, 2006.
- [33] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012.
- [34] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [35] F. Zhan, Y. Yu, K. Cui, G. Zhang, S. Lu, J. Pan, C. Zhang, F. Ma, X. Xie, and C. Miao, “Unbalanced feature transport for exemplar-based image translation,” 06 2021, pp. 15 023–15 033.
- [36] S.-C. Lam, “Texture feature extraction using gray level gradient based co-occurrence matrices,” in *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929)*, vol. 1, 1996, pp. 267–271 vol.1.
- [37] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223.
- [38] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *Int. J. Comput. Vision*, vol. 40, no. 2, p. 99–121, nov 2000.
- [39] P. Diaconis and S. L. Zabell, “Updating subjective probability,” *Journal of the American Statistical Association*, vol. 77, no. 380, pp. 822–830, 1982.
- [40] S. M. Lee, J. H. Xin, and S. Westland, “Evaluation of image similarity by histogram intersection,” *Color Research & Application*, vol. 30, no. 4, pp. 265–274, 2005.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2015.
- [42] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition.”
- [43] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” *CoRR*, vol. abs/1603.08155, 2016.
- [44] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” 06 2015, pp. 3431–3440.
- [45] N. H. T. S. A. U.S. Department of Transportation, “Pre-crash scenario typology for crash avoidance research,” 2007.
- [46] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *International Journal of Robotics Research (IJRR)*, 2013.
- [47] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, “Virtual worlds as proxy for multi-object tracking analysis,” in *CVPR*, 2016.
- [48] B. Wu, A. Wan, X. Yue, and K. Keutzer, “Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3d lidar point cloud,” *CoRR*, vol. abs/1710.07368, 2017. [Online]. Available: <http://arxiv.org/abs/1710.07368>
- [49] Z. Li, X. Wang, X. Liu, and J. Jiang, “Binsformer: Revisiting adaptive bins for monocular depth estimation,” 2022.
- [50] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [51] “Unity3d.” <https://unity.com>, 2021.
- [52] “Donkey Car,” <https://www.donkeycar.com/>, 2021.

- [53] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.
- [54] D. Wu, M. Liao, W. Zhang, and X. Wang, "YOLOP: you only look once for panoptic driving perception," *CoRR*, vol. abs/2108.11250, 2021. [Online]. Available: <https://arxiv.org/abs/2108.11250>
- [55] A. Sarda, S. Dixit, and A. Bhan, "Object detection for autonomous driving using yolo algorithm," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEIM)*, 2021, pp. 447–451.
- [56] P. Ruan, "object-detection—yolov3," <https://github.com/patrick013/Object-Detection---Yolov3/tree/master>, 2020.
- [57] openvino A.I., "Yolo-v3-tf[]," 2020. [Online]. Available: https://docs.openvino.ai/2023.1/omz_models_model_yolo_v3_tf.html
- [58] U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, no. 2, 2022.
- [59] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.
- [60] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: <http://arxiv.org/abs/1604.07316>
- [61] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of ICSE '18*, ser. ICSE '18. New York, NY, USA: ACM, 2018, pp. 303–314. [Online]. Available: <https://doi.acm.org/10.1145/3180155.3180220>
- [62] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of Real Faults in Deep Learning Systems," ser. ICSE'20. New York, NY, USA: ACM, 2020.
- [63] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *CoRR*, vol. abs/1704.07911, 2017.
- [64] G. Jahangirova, A. Stocco, and P. Tonella, "Quality metrics and oracles for autonomous vehicles testing," in *Proceedings of 14th IEEE International Conference on Software Testing, Verification and Validation*, ser. ICST '21. IEEE, 2021.
- [65] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1814–1821.
- [66] M. Weiss and P. Tonella, "Fail-safe execution of deep learning based systems through uncertainty monitoring," in *IEEE 14th International Conference on Software Testing, Validation and Verification*, ser. ICST '21. IEEE, 2021.
- [67] R. Grewal, P. Tonella, and A. Stocco, "Predicting Safety Misbehaviours in Autonomous Driving Systems using Uncertainty Quantification," in *Proceedings of 17th IEEE International Conference on Software Testing, Verification and Validation*, ser. ICST '24. IEEE, 2024, p. 12 pages.
- [68] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML '16. JMLR.org, 2016.
- [69] A. Stocco, P. J. Nunes, M. d'Amorim, and P. Tonella, "ThirdEye: Attention Maps for Safe Autonomous Driving Systems," in *Proceedings of 37th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '22. IEEE/ACM, 2022.
- [70] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016.
- [71] N. P. Jouppi, G. Kurian, S. Li, P. Ma, R. Nagarajan, L. Nai, N. Patil, S. Subramanian, A. Swing, B. Towles, C. Young, X. Zhou, Z. Zhou, and D. Patterson, "Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings," 2023.
- [72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [73] W. Kirch, Ed., *Pearson's Correlation Coefficient*. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091. [Online]. Available: https://doi.org/10.1007/978-1-4020-5614-7_2569
- [74] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: <http://arxiv.org/abs/1512.00567>
- [75] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
- [76] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *CoRR*, vol. abs/2105.15203, 2021.
- [77] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [78] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.
- [79] J. Cohen, *Statistical power analysis for the behavioral sciences*. Hillsdale, NJ: L. Erlbaum Associates, 1988.
- [80] Burch, Najm, Yang, and Trick, "Mcpower: a monte carlo approach to power estimation," in *1992 IEEE/ACM International Conference on Computer-Aided Design*, 1992, pp. 90–97.
- [81] F. U. Haq, D. Shin, S. Nejati, and L. Briand, "Comparing offline and online testing of deep neural networks: An autonomous car case study," in *Proceedings of 13th IEEE International Conference on Software Testing, Verification and Validation*, ser. ICST '20. IEEE, 2020.
- [82] "Replication package," <https://github.com/ast-fortiss-tum/I2I-quality-metrics-study.git>, 2023.
- [83] S. García, D. Strüber, D. Brugali, T. Berger, and P. Pelliccione, "Robotics Software Engineering: A Perspective from the Service Robotics Domain," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020. USA: Association for Computing Machinery, 2020, p. 593–604.
- [84] X. Hu, S. Li, T. Huang, B. Tang, and L. Chen, "Sim2real and digital twins in autonomous driving: A survey," *arXiv preprint arXiv:2305.01263*, 2023.
- [85] C. Schwarz and Z. Wang, "The role of digital twins in connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 41–51, 2022.
- [86] A. Kalapos, C. Górr, R. Moni, and I. Harmati, "Sim-to-real reinforcement learning applied to end-to-end vehicle control," in *2020 23rd International Symposium on Measurement and Control in Robotics (ISMCR)*. IEEE, 2020, pp. 1–6.
- [87] H. Zhou, W. Li, Y. Zhu, Y. Zhang, B. Yu, L. Zhang, and C. Liu, "Deepbillboard: Systematic physical-world testing of autonomous driving systems," 2018.
- [88] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, "Understanding domain randomization for sim-to-real transfer," *CoRR*, vol. abs/2110.03239, 2021.
- [89] H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, "Review of hardware-in-the-loop simulation and its prospects in the automotive area," in *Modeling and simulation for military applications*, vol. 6228. SPIE, 2006, pp. 117–136.
- [90] D. Parthasarathy and A. Johansson, "SilGAN: Generating driving maneuvers for scenario-based software-in-the-loop testing," *CoRR*, vol. abs/2107.07364, 2021.
- [91] Z. Kong and C. Liu, "Generating Adversarial Fragments with Adversarial Networks for Physical-world Implementation," *CoRR*, vol. abs/1907.04449, 2019.
- [92] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretschmar, "SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving," *CoRR*, vol. abs/2005.03844, 2020.
- [93] S. Arora and Y. Zhang, "Do gans actually learn the distribution? an empirical study," *CoRR*, vol. abs/1706.08224, 2017.
- [94] G. Huang, Y. Yuan, Q. Xu, C. Guo, Y. Sun, F. Wu, and K. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," 2018.
- [95] A. Stocco, M. Weiss, M. Calzana, and P. Tonella, "Misbehaviour prediction for autonomous driving systems," in *Proceedings of 42nd International Conference on Software Engineering*, ser. ICSE '20. ACM, 2020, p. 12 pages.
- [96] A. Stocco and P. Tonella, "Towards anomaly detectors that learn continuously," in *Proceedings of 31st International Symposium on Software Reliability Engineering Workshops*, ser. ISSREW 2020. IEEE, 2020.
- [97] ———, "Confidence-driven weighted retraining for predicting safety-critical failures in autonomous driving systems," *Journal of Software: Evolution and Process*, 2021.